SYSTEM AND METHOD FOR PRINTING DATA RECEIVED FROM AN EXTERNAL CONTENT SOURCE

TECHNICAL FIELD

The present invention relates generally to printing data received from an external content source and, more particularly, to methods and systems that enable faster printing of data received from an external content source.

BACKGROUND

Printing speed is an important feature for printers of all types. Often the speed in pages-per-minute (ppm) is touted as one of the key selling points for a printer. When printing large or computationally-intense files, such as a full page color bitmap image, the time between the user's print request and the ejection of the finished page by the printer may be substantial. This time period is sometimes referred to as "click-to-clunk". When the file to be printed must first be retrieved from an external source, "click-to-clunk" time is often increased significantly.

As an example, assume that a user desires to print a bitmap image that fills an 8 ½ x 11 inch page at 300 dots-per-inch (dpi) in 24 bit color. This uncompressed image file would be over 22 MB in size. Even compressing the file using JPEG compression would still leave a file several MB in size. Further assume that this bitmap image is located on a remote server that the user is accessing over the Internet. The user may view a web page that contains a small representation or icon of the image, commonly referred to as a "thumbnail" of the image.

To print the full image on the user's local printer, the user may select the thumbnail image and execute a print request. At this point a two-step process begins. First, the entire bitmap image is downloaded from the server to the user's system. With a modem operating at 56.6 kbs, this download step could take several minutes. During this time the printer sits idle as it waits to receive data. Only after the entire image is downloaded into memory does the printer begin receiving file data and printing. This transfer delay between

the execution of the print request and the beginning of actual printing can be frustrating to the user, and adds to the overall "click-to-clunk" time. The length of the transfer delay and the overall printing time are dependent upon the data transfer rate between the user system and the server. Where the transfer delay is significant, in addition to experiencing frustration the user may question whether the system and/or printer are operating properly.

SUMMARY OF THE INVENTION

The present invention provides a system and method for printing a data file on a printer. Briefly described, the system comprises a computer program product that includes printing logic stored in a memory and executable by a processor. In a preferred embodiment, the printing logic particularly comprises logic to receive a data file in a stream of data from an external content source, logic to gather a first portion of data from the stream, logic to send the first portion to a printer while continuing to receive the stream, logic to gather a second portion of the data from the stream while the first portion is being printed, and logic to send the second portion to the printer after the first portion is printed. The printing logic may further comprise logic to determine a block size of the portions of data by calculating a data transfer speed and adjusting the block size based on the data transfer speed.

The present invention can also be viewed as providing a method for printing a data file received from a remote content source. In this regard, the method can be broadly summarized by the following steps: receiving a data file in a stream of data from a content source; gathering a first portion of data from the stream; printing the first portion while continuing to receive the stream; gathering subsequent portions of data from the stream; and printing the subsequent portions in order after printing the first portion. The method may further comprise the step of determining a block size of the first portion and/or subsequent portions of data by calculating a data transfer speed and adjusting the block size based on the data transfer speed.

The various embodiments of the present invention disclosed herein address the concern of enabling faster printing of data received from an external content source. Other features and advantages of the present

invention will become apparent to one with skill in the art upon examination of the following drawings and detailed description. It is intended that all such additional features and advantages be included herein within the scope of the present invention.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

The invention can be better understood with reference to the following drawings. The components in the drawings are not necessarily to scale, emphasis instead being placed upon clearly illustrating the principles of the present invention. Moreover, in the drawings, like reference numerals designate corresponding parts throughout the several views.

FIG. 1 is a block diagram of a system for printing a data file on a printer according to the present invention;

FIG. 2 is a flow chart of one embodiment of a method for printing a data file on a printer executed in the system of FIG. 1;

FIG. 3 is a flow chart showing the steps for setting the block size of data based on the connection speed between the content source and the user's system.

DETAILED DESCRIPTION OF THE INVENTION

With reference to FIG. 1, a block diagram of a system 100 for printing a data file on a printer according to one embodiment of the present invention is shown. The system 100 may comprise, for example, a computer system as shown or a dedicated logical circuit that replaces the principle components of the computer system. In the preferred embodiment, the system 100 includes a processor 113 and a memory 116, both of which are electrically coupled to a local interface 119. The local interface 119 may comprise, for example, a data bus with an accompanying control bus as is known by those skilled in the computer art. The local interface 119 provides a conduit for the transfer of data between the various components attached thereto.

The system 100 of Fig. 1 is shown in context with a server 106 and a network 109. The system 100 also comprises a network interface 123 that electrically couples a network 109 to the local interface 119. The network

interface 123 makes data obtained from the server 106 via the network 109 available on the local interface 119. The network interface 123 may include, for example, a modem and an appropriate network card that may be employed to transmit and receive data across the network 109. The network 109 may comprise, for example, the Internet, wide area networks, or other similar networks.

The system 100 also includes one or more output interfaces 126 and one or more input interfaces 129. The output interfaces 126 electrically couple one or more output devices to the local interface 119. Examples of such output devices include a printer 128, a display device 133 and other output devices such as speakers, *etc.* The output interfaces 126 may include, for example, an interface card or other similar device. Likewise, the input interfaces 129 electrically couple one or more input devices to the local interface 119 as shown. The input devices may include, for example, a keyboard 136 or a mouse 139.

The memory 116 may comprise any one of or a combination of a number of memory devices, including both volatile and nonvolatile memory components. Volatile components are those that do not retain data values upon loss of power. Conversely, nonvolatile components retain data upon a loss of power. These volatile and nonvolatile components may include, for example, random access memory (RAM), read-only memory (ROM), hard disk drives, floppy disk drives, compact disk drives, tape drives, and other memory components.

A browser 143 and one or more drivers 145 for communicating with the output devices are stored on the memory 116. Upon execution by the processor 113, the logic of the browser 143 generates a browser graphical user interface that appears on the display device 133. The browser 143 may be employed to display various web pages that are downloaded to the system 100 via the network 109 as known in the art. Also stored on the memory 116 is a plug-in 147 that includes printing logic 149 for controlling the system 100 to print a data file on a printer in accordance with the present invention. As explained in more detail below, in a preferred embodiment the printing logic

149 gathers data received from a remote content source and sends portions of the data to the printer 128 via the driver 145.

With continued reference to Figure 1, the server 106 may include a processor 156 and a memory 159, both of which are electrically coupled to a local interface 163. The local interface 163 may comprise, for example, a data bus with an accompanying control bus as known in the art. The server 106 also includes a network interface 166 that electrically couples the network 109 to the local interface 163, thereby allowing data available from the network 109 to be manipulated by the processor 156 and stored in the memory 159. Also, data from the memory 159 may be transmitted to a remote location on the network 109, such as the system 100, via the network interface 166. For example, a data file 169 stored on the memory 159 may be downloaded from the server 106 to the system 100 via the network 109 and ultimately displayed on the display device 133 or printed on the printer 128.

In an important aspect of the present invention, the system 100 may receive a data file from a remote content source, such as the server 106, and begin printing the data from the file before the entire data file is received. In this manner, the actual printing of the data file begins soon after the print request from the user and is much less dependent upon the modem speed, network traffic and other factors. As shown in Figure 1, the remote content source may comprise the server 106 and the data file may be a full-page image file 169 residing in the memory 159 of the server. A user operating the computer system 100 may view on the display device 133 a web page 146 served from server 106. The web page 146 may include a thumbnail version 149 of the full-page image file 169. If the user desires to print the complete image file 169, the user may select the thumbnail version 149 and execute a print request.

Turning now to Figure 2, a flow chart of one embodiment of the system and method of the present invention is illustrated. Beginning with block 200, when the print request is executed the printing logic 149 controls the system 100 to begin receiving the image file 169 from the server 106 in the form of a data stream through the browser 143. In block 202 the printing logic 149

gathers a first portion of data 40' from the stream and stores the first portion in a temporary storage segment 40 of the memory 116 (block 204). The printing logic 149 then progresses to block 206 and sends the first portion of data 40' to the printer 128 via the driver 145. Upon receiving the first portion of data 40', the printer 128 begins printing. Meanwhile, in block 208 the printing logic 149 gathers a second portion of data 40" from the stream and stores the second portion in the memory 116 (block 210). It will be appreciated that the step of gathering a second portion of data 40" (block 208) may be performed concurrently with the step of sending the first portion of data 40' to the printer (block 206), or may be started while the first portion of data 40' is still printing. Additionally, this step and the other steps described herein are performed in the background such that the user is unaware that data is being continuously received.

After storing the second portion of data 40" in memory (block 210), the printing logic 149 then proceeds to block 212 to determine if the printer has finished printing the first portion of data 40'. If the printer has finished printing the first portion of data 40', then the printing logic 149 sends the second portion of data 40" to the printer 128 (block 214). If the printer has not finished printing the first portion of data 40', the printing logic 149 progresses to block 216 and determines whether a printing timeout has expired, where the printing timeout comprises a predefined period of time. When the printing timeout expires, the printing logic 149 again executes the query of block 212.

It will be appreciated that the block size of the first, second and any additional portions of data may be a predefined value or may be adjusted based upon one or more factors, such as data transfer speed between the computer system 100 and the server 106. A more detailed explanation of one embodiment of logic that adjusts the block size of the data portions is provided below.

After storing the second portion of data 40" in memory, the printing logic 149 also determines whether the entire data file 169 has been received from the server 106. More specifically, in block 220 of the illustrated embodiment the printing logic 149 determines whether a "destroy stream" command has been received. If a "destroy stream" command has been

received, the printing logic 149 ends. If a "destroy stream" command has not been received, the printing logic 149 progresses to block 222 and gathers the Next portion of data from the stream. The printing logic 149 then stores the Next portion in the temporary storage segment 40 of the memory 116 (block 224). The printing logic 149 then progresses to block 226 where it determines if the printer 128 has finished printing the Previous portion of data. If the printer has finished printing the Previous portion, then the printing logic 149 sends the Next portion of data to the printer 128 (block 228). If the printer has not finished printing the Previous portion of data, the printing logic 149 progresses to block 230 and determines whether a printing timeout has expired, where the printing timeout comprises a predefined period of time. When the printing timeout expires, the printing logic 149 again executes the query of block 226. Additionally, after storing the Next portion of data in memory at block 224, the printing logic 149 returns to block 220 to determine if a "destroy stream" command has been received, and the subsequent steps described above are repeated as appropriate.

With reference now to Figure 3, a flow chart of another advantageous feature of one embodiment of the printing logic 149 is illustrated. More specifically, Figure 3 shows the steps executed by the printing logic 149 to set a block size for the portions of data gathered from the data stream. Preferably, the steps of Figure 3 are executed before the system 100 begins receiving data from the remote content source at block 200 in Figure 2. The steps of Figure 3 may also be executed periodically and concurrently with the printing process described in Figure 2. In this manner, the block size of the portions of data may be dynamically adjusted during the printing process to optimize print speed.

Beginning with block 240 in Figure 3, in a preferred embodiment the printing logic 149 pings the server 106 and calculates a data transfer speed between the system 100 and the server (block 242). The printing logic 149 then proceeds to block 244 and determines if the data transfer speed is greater than a predetermined threshold value A, such as 28.8 kbs. If the data transfer speed is not greater than A, then the block size is set to a predetermined value W, such as 4KB (block 246). If the data transfer speed

is greater than A, then the printing logic 149 proceeds to block 248 where it determines if the data transfer speed is greater than another predetermined threshold value B, such as 56 kbs. If the data transfer speed is not greater than B, then the block size is set to another predetermined value X (block 250), such as 16KB, where X>W. If the data transfer speed is greater than B, then the printing logic 149 proceeds to block 252 and sets the block size to another predetermined value Y, such as 64KB, where Y>X. Thereafter, this portion of the printing logic 149 ends accordingly.

Advantageously, by increasing the block size of data as the data transfer speed increases, fewer portions of data are sent to the printer. It will be appreciated that the printing logic 149 may utilize only one comparison of the data transfer rate to a single predetermined threshold value. Similarly, three or more such comparisons to different threshold values may also be utilized. It will also be appreciated that system characteristics other than data transfer speed may be examined and utilized to adjust the block size of data. These characteristics may include printer capabilities, printer performance, processor speed, etc.

Referring back to Figure 1, and in an alternative embodiment of the system and method of the present invention, at least a portion of the printing logic 149 may reside in the memory 159 of the server 106. In this manner, one or more of the functions described in the flow charts of Figures 2 and 3 may be performed by the server. For example, upon receiving a request from the system 100 to download the data file 169, the server 106 may partition the data file 169 into a plurality of portions. The server 106 may then transfer a first portion of the plurality of portions of data to the system 100 via the network 109, and the system may begin printing the first portion on the printer 128. The server 106 may then transfer a second portion of the plurality of portions of data to the system 100, and the system may print the second portion after printing the first portion. This process may continue until the entire data file 169 has been transferred to the system 100.

As described above, the various embodiments of the printing logic 149 provide a distinct advantage in that a printer may begin printing a large data file without waiting for the entire file to be received from a remote content

source. Also, the various embodiments of the printing logic 149 described above may include logic to adjust the block size of data portions sent to the printer according to one or more factors, such as the data transfer rate.

The printing logic 149 of the present invention may be implemented in hardware, software, firmware, or a combination thereof. In the preferred embodiment(s), the printing logic 149 is implemented in software or firmware that is stored in a memory and that is executed by a suitable instruction execution system. If implemented in hardware, as in an alternative embodiment, the printing logic 149 can implemented with any or a combination of the following technologies, which are all well known in the art: a discrete logic circuit(s) having logic gates for implementing logic functions upon data signals, an application specific integrated circuit having appropriate logic gates, a programmable gate array(s) (PGA), a fully programmable gate array (FPGA), *etc.*

The flow charts of Figures 2 and 3 show the architecture, functionality, and operation of possible implementations of the printing logic 149. In this regard, each block represents a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). Note that in some alternative implementations, the functions contained in the blocks may occur out of the order noted in Figures 2 and 3. For example, two blocks shown in succession in Figures 2 and 3 may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved.

The printing logic 149, which comprises an ordered listing of executable instructions for implementing logical functions, can be embodied in any computer-readable medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In the context of this document, a "computer-readable medium" can be any means that can contain, store, communicate, propagate or transport the program for use by or in connection with the

instruction execution system, apparatus, or device. The computer readable medium can be, but is not limited to, an electronic, magnetic, optical, electromagnetic, infrared or semiconductor system, apparatus, device or propagation medium. More specific examples (a nonexhaustive list) of the computer-readable medium would include the following: an electrical connection (electronic) having one or more wires, a portable computer diskette (magnetic), a random access memory (RAM) (magnetic), a read-only memory (ROM) (magnetic), an erasable programmable read-only memory (EPROM or Flash memory) (magnetic), an optical fiber (optical) and a portable compact disc read-only memory (CDROM) (optical). Note that the computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via for instance optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.

Many variations and modifications may be made to the above-described embodiment(s) of the invention without departing substantially from the spirit and principles of the invention. All such modifications and variations are intended to be included herein within the scope of the present invention.